OIPE JC184
AUG 2 6 2005
PATENT & TRADEMARK OFFICE

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission: 17

Application Number	10/016,700
Filing Date	11/02/2001
First Named Inventor	Challener et al.
Art Unit	2161
Examiner Name	Cindy Nguyen
Attorney Docket Number	RPS920000400US2

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☑ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☑ Other Enclosure(s) (please Identify below):

Request for Reinstatement of Appeal; Return Postcard

Remarks

Supplemental Appeal Brief

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Winstead Sechrest & Minick P.C.		
Signature			
Printed name	Kelly K. Kordzik		
Date	08/24/2005	Reg. No.	36,571

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Toni Stanley		
Typed or printed name	Toni Stanley	Date	08/24/2005

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

AUG 26 2005
PATENT & TRADEMARK OFFICE

AF/2161
BFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Challener et al.	:	Before the Examiner: Cindy Nguyen
Serial No.: 10/016,700	:	Group Art Unit: 2161
Filed: November 2, 2001	:	
Title: TRUSTED COMPUTING PLATFORM WITH DUAL KEY TREES TO SUPPORT MULTIPLE PUBLIC PRIVATE KEY SYSTEMS	:	Lenovo, (United States) Inc. Building 675 4401 Silicon Drive Durham, NC 27709

REQUEST FOR REINSTATEMENT OF APPEAL

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office Action having a mailing date of June 16, 2005, reopening prosecution of the above-referenced Application, Applicants respectfully request reinstatement of the Appeal based on the Appeal Brief filed on April 21, 2005, and the Notice of Appeal filed on February 17, 2005. A Supplemental Appeal Brief is filed herewith.

CERTIFICATION UNDER 37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on August 24, 2005.

Toni Stanley
Signature

Toni Stanley
(Printed name of person certifying)

FEE DEFICIENCY

NOTE: If there is a fee deficiency and there is no authorization to charge an account, additional fees are necessary to cover the additional time consumed in making up the original deficiency. If the maximum, six-month period has expired before the deficiency is noted and corrected, the application is held abandoned. In those instances where authorization to charge is included, processing delays are encountered in returning the papers to the PTO Finance Branch in order to apply these charges prior to action on the cases. Authorization to charge the deposit account for any fee deficiency should be checked. See the Notice of April 7, 1986, 1065 O.G. 31-33.

☒ Applicants believe no fee is due; however, if any additional extension and/or fee is required, this is a request therefor and to charge Account No. 50-3533 (RPS920000400US2).

AND/OR

☒ If any additional fee for claims is required, charge Account No. 50-3533 (RPS920000400US2).

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Attorneys for Applicants

By: _____

Kelly K. Kordzik
Reg. No. 36,571

P.O. Box 50784
Dallas, Texas 75201
(512) 370-2851

Austin_1 289535v.1

OIPE JC164
AUG 26 2005
PATENT & TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Challener et al.	:	Before the Examiner: Cindy Nguyen
Serial No.: 10/016,700	:	Group Art Unit: 2161
Filed: November 2, 2001	:	
Title: TRUSTED COMPUTING PLATFORM WITH DUAL KEY TREES TO SUPPORT MULTIPLE PUBLIC/PRIVATE KEY SYSTEMS	:	Lenovo, (United States) Inc. Building 675 4401 Silicon Drive Durham, NC 27709

SUPPLEMENTAL APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, which is the assignee of the entire right, title and interest in the above-identified patent application.

CERTIFICATION UNDER 37 C.F.R. §1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on 8-24, 2005.

Toni Stanley
Signature

Toni Stanley
(Printed name of person certifying)

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, Appellants' legal representative or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-8, 18-24 and 27 are pending in the application. Claims 9-17 have been allowed. Claims 1-6, 8, 18-25 and 27 stand rejected. Claims 7 and 26 are objected to.

IV. STATUS OF AMENDMENTS

Applicants have filed an Amendment After Final amending claims 7 and 26 to be in independent form.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Referring to FIGURE 5, there is illustrated an embodiment of the present invention for the creation and use of keys within a TPM, such as TPM 951 described below with respect to FIGURE 9. There is a TPM storage root key 501 and a platform migratable key 502. Additionally, there are user migratable keys 503 and 504 and signing keys 507 thereunder. All such keys are migratable. The present invention, however, makes use of the ability of a TPM to have non-migratable keys as well as migratable keys. Migratable keys can be transferred to other TPMs, and non-migratable keys cannot be transferred. Thus, such non-migratable keys are locked to the hardware, i.e., the TPM 951. With such non-migratable keys, the TPM 951 can only decrypt such keys. [Page 8, lines 10-19]

Such migratable and non-migratable keys are desired within the TPM, but the use of deeply embedded migratable keys is not desired because it takes too long for such embedded key structures to load. But yet, it is desirable to have such keys migratable for maintenance purposes, such as to move a single user from one platform to another or to move an entire platform from one system to another. It is not desirable in such instances to go through the system and find every single key, to determine what kind of key it is and then migrate such keys individually. [Page 8, line 20 – page 9, line 4]

As noted above, deeply embedded trees of keys take a relatively long time to load. For example, if it is desired to have a signing key, then that signing key will be encrypted with a public key of a user key, which may be encrypted with the key of a department, which may be encrypted with key of the platform which is encrypted with the storage root key. All such keys within the tree need to be loaded. However, since it is not desirable that every member of a department have access to the keys of every user in that department, individual user authentication data may be associated with each user key, so that only the appropriate user is allowed to load keys associated with that user. This is especially the case as a given "leaf" key may be set to not use user authentication data in order to be used, so loading the key in this case is equivalent to being able to use the key. Ease of use and security constraints dictate that there not be two sets of user authentication data for loading a key. [Page 9, lines 5-16]

Referring to FIGURE 6, in the present invention, in step 601, a new migratable signing key is created. Then, in step 602, the new migratable signing key is stored in the user migratable storage key 503 or 504. In step 603, the new migratable signing key is also stored in the user non-migratable storage key 505 or 506. By design, the same user authentication data is used to perform this action for both storage keys, so the user only needs to provide it once. Since the user non-

migratable storage key 505 or 506 is the faster type of public/private key, it will load faster when the migratable signing key is needed. [Page 10, lines 3-10]

In a similar way, when a new migratable storage key is requested to be created and stored under a specified migratable storage key M, the request will be translated into two requests. The first will behave exactly as specified by the TCPA specification. The second request will request a non-migratable storage key (of faster type) to be created and stored under the non-migratable storage key corresponding to M in the fast tree. Both requests will contain the same user authorization data, and then the database on the system which associates migratable storage keys and non-migratable storage keys will be updated to reflect the new correspondence between the two newly created keys. [Page 10, lines 11-19]

Referring to FIGURE 7, in step 701, a request for a signature by a key is made. In step 702, the database is searched for the location of the key blob to load. In step 703, a copy of the key stored in the non-migratable storage key blob is loaded, and in step 704, the key is used to execute the signature. [Page 11, lines 7-10]

In the present invention, in FIGURE 8, in step 801, a migration of a key is requested. In step 802, the database is searched for the location of the key blob to load. In step 803, a copy of the key stored in the non-migratable storage key blob is loaded, and this key is used to sign in step 804. The present invention allows users to store and load keys much more quickly with faster public/private keys than 2048 bit RSA keys. However, the present invention preserves both the ability to migrate keys and also the structure of user authentication data needed to load or use a key. [Page 12, lines 11-19]

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-6, 18-25 and 27 stand rejected under 35 U.S.C. § 102(e) as being anticipated by *Ober et al.* (U.S. Patent No. 6,307,936).

VII. ARGUMENT

Claims 1-6, 8, 18-25 and 27 are not properly rejected under 35 U.S.C. § 102(e) as being anticipated by *Ober*. As the Examiner is well aware, for a claim to be anticipated under § 102, each and every element of the claim must be found within the cited prior art reference. As Applicant will hereinafter assert, the Examiner has not sufficiently shown where in *Ober* the various limitations of the claims are found. Though an examiner may give claim language a broad interpretation, such an interpretation must be reasonable, and also consistent with the interpretation that those skilled in the art would reach, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in Applicant's Specification. MPEP § 2111.

A migratable key can be transferred to other trusted platform modules chips, while non-migratable keys cannot be transferred. Specification, page 8, lines 16-17. Thus, such non-migratable keys are locked to the hardware, which is the only hardware that can decrypt such keys. Specification, page 8, lines 17-19.

Claims 1 and 18

Regarding Claims 1 and 18, the Examiner asserts that *Ober* discloses creating a migratable storage tree with a storage root key in column 5, line 59 through column 6, line 12. Applicants respectfully traverse. Nowhere within this cited language is this limitation found. Further, this cited language in *Ober* does not even include the words "migratable," "storage tree," or "storage root key." If the Examiner believes these claim limitations are taught within this cited language, then the Examiner cannot merely recite the "creating step" of the claims and point to a couple of

paragraphs and merely assert that this claim limitation is taught within those paragraphs without further explaining how they are taught. In fact, the Examiner has done this repeatedly with every rejection in the office action.

Column 6, lines 25-39 of *Ober* does not disclose the step of creating a non-migratable storage tree with the storage root key. In fact, the term "non-migratable storage tree" is not even found anywhere within this language. Neither is the entire term "storage root key."

Column 6, line 65 through column 7, line 8 does not disclose wherein the migratable storage tree and the non-migratable storage tree are identically structured with corresponding keys and authentication data. In fact, the terms "migratable storage tree," "non-migratable storage tree," and "authentication data" are not even found within this cited language.

Claims 2 and 19

Claims 2 and 19 recite that the migratable storage tree and the non-migratable storage tree are created by a trusted computing module in accordance with the Trusted Computing Platform Alliance. The Examiner has attempted to reject these claims by citing column 6, lines 49-65 of *Ober*. There is absolutely no disclosure of a trusted computing module or the Trusted Computing Platform Alliance in *Ober*. Language referring to trusted or untrusted keys does not anticipate these specific claim limitations.

Claims 3 and 20

With respect to claims 3 and 20, these claims recite that the migratable storage tree comprises migratable keys and a user key, wherein the non-migratable storage tree comprises non-migratable keys and a user key. The Examiner has attempted to reject these claims by citing column 6, lines 1-39 of *Ober*. The Examiner has failed to specifically point out in this language where these limitations are found, and

instead has made a blanket assertion using Applicants' claim language, but not citing to any specific *Ober* language. This language in *Ober* does not anywhere make any reference to "migratable storage tree," "migratable keys," "non-migratable storage tree," or "non-migratable keys."

Claims 4 and 22

With respect to claims 4 and 22, these claims recite that the non-migratable storage tree will include non-migratable storage keys corresponding to each migratable storage key in the migratable storage tree. In rejecting these claims, the Examiner has recited column 6, line 49 through column 7, line 8 of *Ober*. There is absolutely no teaching or inference of non-migratable storage trees having non-migratable storage keys corresponding to migratable storage keys in a migratable storage tree. Not only is the Examiner merely citing a multiple of paragraphs within *Ober*, but the Examiner has not specifically pointed out which language within the cited paragraphs particularly pertains to the claim limitations.

Claims 5 and 24

With respect to claims 5 and 24, these claims recite that use authorization in the non-migratable storage tree will be identical to use authorization in the migratable storage tree. The Examiner has attempted to reject these claims by referring to column 7, lines 51-65 of the *Ober* reference. This language within *Ober* merely discusses supporting trusted or untrusted key storage. However, this language does not specifically disclose that use authorization in the non-migratable storage tree will be identical to use authorization in the migratable storage tree.

Claim 6

Claim 6 recites the further steps of requesting a migratable storage tree and requesting a non-migratable storage key. The cited language of *Ober* does not even

remotely disclose these limitations. Again, the examiner needs to show how the language cited by the examiner in *Ober* matches up with the language in the claims.

Claim 8

The Examiner has not in any way addressed the claim limitations of claim 8. In fact, the Examiner has not even mentioned claim 8 within any of the Examiner's rejections. For this reason alone, the Examiner has failed to prove a *prima facia* case of anticipation in rejecting claim 8.

Claim 21

With respect to claim 21, column 5, lines 44-67 of *Ober* do not in any way disclose these limitations, and the Examiner has failed to show how they do.

Claim 23

With respect to claim 23, again the Examiner has merely cited language in column 6, lines 25-36 of *Ober* without specifically pointing out how this language matches up with the claim limitations. This language within *Ober* does not even mention the terms "non-migratable storage tree," "non-migratable storage keys," "migratable storage keys," or "migratable storage tree."

Claims 25 and 27

Column 6, lines 1-12 of *Ober* does not in any way discuss "non-migratable storage trees," or "migratable storage trees," or that non-migratable storage trees can be deduced from user authorization in the migratable storage tree with additional data. With respect to claim 27, column 5, lines 22-29 of *Ober* does not in any way teach or suggest a migratable key can be transferred to other trusted platform module chips in where a non-migratable key cannot be transferred to other trusted platform module chips.

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Attorneys for Appellants

By: _____

Kelly K. Kordzik

Reg. No. 36,571

P.O. Box 50784
Dallas, Texas 75201
(512) 370-2832

APPENDIX

1. In a data processing system, a method comprising the steps of:

creating a migratable storage tree with a storage root key; and

creating a non-migratable storage tree with the storage root key, wherein the migratable storage tree and the non-migratable storage tree are identically structured.

2. The method as recited in claim 1, wherein the migratable storage tree and the non-migratable storage tree are created by a trusted computing module in accordance with Trusted Computing Platform Alliance.

3. The method as recited in claim 1, wherein the migratable storage tree comprises migratable keys and a user key, wherein the non-migratable storage tree comprises non-migratable keys and a user key.

4. The method as recited in claim 1, wherein the non-migratable storage tree will include non-migratable storage keys corresponding to each migratable storage key in the migratable storage tree.

5. The method as recited in claim 1, wherein use authorization in the non-migratable storage tree will be identical to use authorization in the migratable storage tree.

6. The method as recited in claim 1, further comprising the steps of:

requesting a migratable storage key; and

requesting a non-migratable storage key.

7. The method as recited in claim 6, wherein the step of requesting a migratable storage key will identify a parent key in the migratable storage tree, and wherein the

step of requesting a non-migratable storage key will identify a parent key in the non-migratable storage tree that corresponds to the parent key in the migratable storage tree.

8. The method as recited in claim 1, further comprising the step of:

when a key loading request is made for a migratable storage key, loading a key from the non-migratable storage tree instead of loading a corresponding key from the migratable storage tree.

9. In a data processing system, a method comprising the steps of:

splitting a request to create a new migratable storage key with given authentication data and a first parent key into first and second commands;

wherein the first command creates a migratable storage key with the given authentication data and the first parent key; and

wherein the second command requests creating a non-migratable storage key with the given authentication data and a second parent key which is determined from looking up a key that corresponds to the first parent key in a database.

10. The method recited in claim 9, wherein the migratable storage key and the non-migratable storage key are associated in a database.

11. The method recited in claim 9, wherein the non-migratable key is a multi-prime key.

12. The method recited in claim 9, where the non-migratable key is an elliptic curve key.

13. The method as recited in claim 9, further comprising the steps of:

creating a new migratable signing key with the given authentication data and a third parent key;

storing the new migratable signing key with the given authentication data and the third parent key;

storing the new migratable signing key with the given authentication data and a fourth parent key where the fourth parent key is a non-migratable key associated with the third parent key in a database.

14. The method as recited in claim 13, further comprising the steps of:

requesting a signature by the new migratable signing key;

searching the database for the location of a key blob containing the new migratable signing key;

loading a copy of the new migratable signing key stored in the key blob created with the non-migratable parent key; and

signing with the new migratable signing key.

15. The method as recited in claim 9, further comprising the steps of:

creating a new data stored by means of the first parent key;

storing the new data with the first parent key;

storing the new data with the second parent key where the second parent key is a non-migratable key associated with the third parent key in a database.

16. The method as recited in claim 15, further comprising the steps of:

requesting data stored by the new migratable storage key;

searching the database for the location of a key blob associated with the new migratable storage key;

loading a copy of the key blob created with the non-migratable storage key; and decrypting the data.

17. The method as recited in claim 14, further comprising the steps of:

requesting migration of new migratable signing keys;

searching the database for the location of a key blob associated with a non-migratable parent of the key to be migrated;

processing the migration.

18. In a data processing system, a method comprising the steps of:

creating a migratable storage tree with a storage root key; and

creating a non-migratable storage tree with the storage rootkey where the migratable storage tree and the non-migratable storage tree are identically structured with corresponding keys and authentication data.

19. The method as recited in claim 18, wherein the migratable storage tree and the non-migratable storage tree are created by a trusted computing module in accordance with Trusted Computing Platform Alliance.

20. The method as recited in claim 19, wherein the migratable storage tree comprises migratable keys and a user key, wherein the non-migratable storage tree comprises non-migratable keys and a user key.

21. The method recited in claim 18, wherein the migratable storage tree comprises migratable keys and encrypted user data wherein the non-migratable storage tree comprises non-migratable keys and encrypted user data .

22. The method as recited in claim 18, wherein the non-migratable storage tree will include non-migratable storage keys corresponding to each migratable storage key in the migratable storage tree.

23. The method as recited in claim 18, wherein the non-migratable storage tree will include non-migratable storage keys corresponding to a subset of the migratable storage keys in the migratable storage tree.

24. The method as recited in claim 18, wherein use authorization in the non-migratable storage tree will be identical to use authorization in the migratable storage tree.

25. The method as recited in claim 18, wherein use authorization in the non-migratable storage tree can be deduced from user authorization in the migratable storage tree with additional data.

26. The method as recited in claim 25, wherein the use authorization in the non-migratable storage tree is obtained by hashing the concatenation of the user authorization in the migratable storage tree with a fixed string.

27. The method as recited in claim 1, wherein a migratable key can be transferred to other trusted platform module chips, and wherein a non-migratable key cannot be transferred to other trusted platform module chips.